IN THE CLAIMS:

1. (Currently Amended) A tangible machine-accessible medium including data that, when accessed by one or more machines, causes said one or more machines to:

multiply-add a first line of packed byte data with a first line of packed transform coefficients to generated a first intermediate packed data including a first sum of products and a second sum of products; and

horizontal-add the first sum of products and the second sum of products to generate a ~~to generate a~~ first result of a first plurality of packed results.

2. (Original) The machine-accessible medium of claim 1 further including data that, when accessed by said one or more machines, causes said one or more machines to:

multiply-add a second line of packed byte data with the first line of packed transform coefficients to generated a second intermediate packed data including a third sum of products and a fourth sum of products; and

horizontal-add the third sum of products and the fourth sum of products to generate a to generate a second result of the first plurality of packed results.

3. (Original) The machine-accessible medium of claim 1 further including data that, when accessed by said one or more machines, causes said one or more machines to:

multiply-add the first line of packed byte data with a second line of packed transform coefficients to generated a second intermediate packed data including a third sum of products and a fourth sum of products; and

horizontal-add the third sum of products and the fourth sum of products to generate a to generate a second result of the first plurality of packed results.

4. (Original) The machine-accessible medium of claim 1 including data that, when

accessed by said one or more machines, causes said one or more machines to treat elements of the first line of packed byte data as unsigned bytes in generating the first and second sums of products of the first intermediate packed data.

5. (Original) The machine-accessible medium of claim 4 including data that, when accessed by said one or more machines, causes said one or more machines to treat elements of the first line of packed transform coefficients as signed bytes in generating the first and second sums of products of the first intermediate packed data.

6. (Original) The machine-accessible medium of claim 5 including data that, when accessed by said one or more machines, causes said one or more machines to treat elements of the first intermediate packed data as signed 16-bit words in generating the first result and to horizontal-add the first and second sums of products using saturation.

7. (Original) The machine-accessible medium of claim 1 including data that, when accessed by said one or more machines, causes said one or more machines to overwrite the first line of packed byte data with the second intermediate packed data and to overwrite the second intermediate packed data with the first plurality of packed results.

8. (Original) A method comprising:

decoding a first multiply-add instruction, a second multiply-add instruction and a horizontal-add instruction, each of an instruction format to specify a first operand and a second operand;

responsive to said first multiply-add instruction, wherein said first multiply-add instruction specifies a first packed data including a_1, a_2, a_3 and a_4 byte data elements, and a second packed data including b_1, b_2, b_3 and b_4 byte data elements, performing an operation $(a_1 \times b_1) + (a_2 \times b_2)$ to generate a 16-bit data element e_1 of a third packed

data, performing an operation $(a_3 \times b_3) + (a_4 \times b_4)$ to generate a 16-bit data element e_2 of the third packed data;

responsive to said second multiply-add instruction, wherein said second multiply-add instruction specifies a fourth packed data including c_1, c_2, c_3 and c_4 byte data elements, and a fifth packed data including d_1, d_2, d_3 and d_4 byte data elements, performing an operation $(c_1 \times d_1) + (c_2 \times d_2)$ to generate a 16-bit data element f_1 of a sixth packed data, performing an operation $(c_3 \times d_3) + (c_4 \times d_4)$ to generate a 16-bit data element f_2 of the sixth packed data; and

responsive to said horizontal-add instruction, wherein said horizontal-add instruction specifies the third packed data and the sixth packed data, performing an operation $(e_1 + e_2)$ to generate a data element g_1 of a seventh packed data, performing an operation $(f_1 + f_2)$ to generate a data element g_2 of the seventh packed data.

9. (Original) The method of claim 8 wherein elements of the first and fourth packed data are treated as unsigned bytes.

10. (Original) The method of claim 9 wherein elements of the second and fifth packed data are treated as signed bytes.

11. (Original) A method comprising:

decoding a multiply-add instruction and an horizontal-add instruction, both of a variable length instruction format comprising at least an opcode field, an addressing mode field, a first and a second source field, said first and second source fields of the multiply-add instruction respectively indicating a first operand having a first plurality of byte data elements including at least A1, A2, A3, and A4 byte data elements, and a second operand having a second plurality of byte data elements including at least B1, B2, B3, and B4 byte data elements;

responsive to decoding said multiply-add instruction, performing the operation (A1 x B1) + (A2 x B2) to generate a first 16-bit data element of a first packed data, and performing the operation (A3 x B3) + (A4 x B4) to generate a second 16-bit data element of the first packed data; and

responsive to said horizontal-add instruction, adding the first and second 16-bit data elements of the first packed data to generate a third 16-bit data element and storing the third 16-bit data element as one of a plurality of data elements of a packed result.

12. (Original) The method of claim 11, said first plurality of byte data elements including at least 16 byte data elements and said second plurality of data elements including at least 16 byte data elements.

13. (Original) The method of claim 11, said first plurality of data elements further including at least A5, A6, A7, and A8 as byte data elements, and said second plurality of data elements further including at least B5, B6, B7, and B8 as data elements, the method further comprising:

responsive to said second opcode field, enabling an execution unit with the decoded multiply-add instruction to perform the operation (A5 x B5) + (A6 x B6) to generate a third 16-bit data element of the packed result data, and to perform the operation (A7 x B7) + (A8 x B8) to generate a fourth 16-bit data element of the packed result data.

14. (Original) The method of claim 11 wherein said first plurality of data elements are treated as unsigned bytes.

15. (Original) The method of claim 14 wherein said second plurality of data elements are treated as signed bytes.

16. (Original) The method of claim 15 wherein each of said first, second, third and fourth 16-bit data elements are generated using signed saturation.

17. (Original) An apparatus to perform the method of claim 16 comprising:

a packed multiply-adder circuit;

at least one state machine, and

a machine-accessible medium including data that, when accessed by said at least one state machine, causes said at least one state machine enable the packed multiply-adder circuit to perform the method of Claim 16.

18. (Original) An apparatus to perform the method of claim 13 comprising:

an operation control unit; and

a machine-accessible medium including data that, when accessed by said operation control unit responsive to said second opcode field, causes the execution unit to perform the method of Claim 13.

19. (Original) The method of claim 13 wherein said first source field comprises bits five through three of the variable length instruction format.

20. (Original) The method of claim 19 wherein said second source field comprises bits two through zero of the instruction format.

21. (Original) The method of claim 20 wherein said first plurality of byte data elements is overwritten by said packed result data responsive to the multiply-add instruction.

22. (Original) An apparatus comprising:

a first circuit to receive a first packed data comprising at least four byte data elements

a second circuit to receive a second packed data comprising at least four byte data elements;

a decoder to decode a plurality of instructions including a first instruction and a second instruction, the first instruction comprising a first source field indicating a first location to access said first packed data, and a second source field indicating a second location to access said second packed data;

a multiply-adder circuit, enabled by the decoded first instruction, to multiply each of a first pair of byte data elements of the first packed data with respective byte data elements of the second packed data and to generate a first 16-bit result representing a first sum of products of the first pair of multiplications, and to multiply each of a second pair of byte data elements of the first packed data with respective byte data elements of the second packed data and to generate a second 16-bit result representing a second sum of products of the second pair of multiplications;

a third circuit to store a third packed data comprising at least said first and second 16-bit results in response to the first instruction;

an adder circuit, enabled by the decoded second instruction, to add said first and second 16-bit results of the third packed data to generate a third 16-bit result representing a third sum of products of the first and second pairs of multiplications; and

a fourth circuit to store a fourth packed data comprising at least said third 16-bit result in response to the second instruction.

23. (Original) The apparatus of claim 22 wherein said first and second packed data each contain at least eight byte data elements.

24. (Original) The apparatus of claim 22 wherein said first and second packed data each

contain at least sixteen byte data elements.

25. (Original) The apparatus of claim 22 wherein the first packed data comprises unsigned byte data elements.

26. (Original) The apparatus of claim 22 wherein the second packed data comprises signed byte data elements.

27. (Original) The apparatus of claim 26 wherein the first packed data comprises unsigned byte data elements.

28. (Original) The apparatus of claim 27 wherein the first and second 16-bit results are generated using signed saturation.

29. (Original) The apparatus of claim 22 wherein the multiply-adder circuit comprises a first and a second 16.times.16 multiplier to perform the first and the second pair of multiplications respectively.

30. (Original) A computing system comprising:

an addressable memory to store data;

a processor including:

a first storage area to store M packed unsigned byte data elements;

a second storage area to store M packed signed byte data elements;

a decoder to decode a first instruction comprising a first opcode field having a hexadecimal value of 0F38, a second opcode field having a hexadecimal value of 04, a first source field indicating said first storage area, and a second source field indicating said second storage area;

an execution unit, responsive to the decoder decoding a first instruction, to produce M products of multiplication of the packed byte data elements stored in the first storage area by corresponding packed byte data elements stored in the second storage area, and to sum the M products of multiplication pairwise to produce M/2 results representing M/2 sums of products; and

a third storage area to store M/2 packed 16-bit data elements, the third storage area corresponding to a destination specified by the first instruction to store the M/2 results; and

a magnetic storage device to store said first instruction.

31. (Original) The computing system of claim 30 wherein M is 16.

32. (Original) The computing system of claim 30 wherein M is 8.

33. (Currently Amended) The computing system of claim 32 ~~23~~ wherein each of said M/2 16-bit results are generated using signed saturation.